Notes for SVM

Shoufu Luo

December 13, 2013

1 Formulation

Linearly Separable In a 2-dimensional space, separate the classes with the largest margin by $\mathbf{w}.\mathbf{x} + b = 0$. For j-th instance, $(\mathbf{w}.\mathbf{x_j} + b) \cdot y_j$ is so-called "confidence", where $y_j \in \{+1, -1\}$. Maximize the margin γ:

$$\dagger \max_{\gamma,w,b} \gamma, \text{ subject to } (\mathbf{w}.\mathbf{x_j} + b) \cdot y_j \geq \gamma, \forall j \in \text{Dataset}$$

Where $2 \cdot \gamma$ is the margin between boundaries of two decision regions. Use Canonical Hyperplanes, $\mathbf{w}.\mathbf{x}^+ + b = +1$ and $\mathbf{x}^+ = \mathbf{x}^- + \lambda \mathbf{w}$, we have $\mathbf{w}.(\mathbf{x}^- + \lambda \frac{\mathbf{w}}{\|\mathbf{w}\|}) + b = +1$. Solve the equalization: $\lambda = \frac{2}{\|\mathbf{w}\|}$ and $\gamma = \frac{1}{\sqrt{\mathbf{w}.\mathbf{w}}}$. Substitute: $\gamma = \frac{1}{\sqrt{\mathbf{w}.\mathbf{w}}}$:
Primal:

$$\dagger \min_{w,b} \mathbf{w}.\mathbf{w}, \text{ subject to } (\mathbf{w}.\mathbf{x_j} + b) \cdot y_j \geq 1, \forall j \in \text{Dataset}$$

Use Lagrange Multipliers α:

$$L(W, \alpha) = \frac{1}{2}\mathbf{w}.\mathbf{w} - \sum_j \alpha_j [(\mathbf{w}.\mathbf{x_j} + b)y_j - 1], \text{ where } \alpha_j \geq 0, \forall j$$

Take the partial w.r.t. to $\mathbf{w}$, α and solve the gradient:

1. $\frac{\partial L}{\partial \mathbf{w}} = 0 \Rightarrow \mathbf{w} = \sum_j \alpha_j y_j x_j$
2. $\frac{\partial L}{\partial \alpha} = 0 \Rightarrow \sum_j \alpha_j y_j = 0$

Dual: Substitute $\mathbf{w}$ with $\sum_j \alpha_j y_j \mathbf{x_j}$ and $\sum_i \alpha_i y_i = 0$

$$\dagger \max_\alpha \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x_i} \mathbf{x_j}, \text{ where } C \geq \alpha_i \geq 0 \ (*)$$

Nonzero α_k define the decision boundaries. The data points $\mathbf{x_i}$ corresponding to nonzero α_k are the support vectors, which gives $b = y_k - \mathbf{w}.\mathbf{x_k}$

Soft Margin Usually, not all data points are perfectly separable. By adding slack variables ξ_j, and a penalty parameters C, an SVM classifier can use a *soft margin*. It means the optimal hyperplane can separate many but not all data points. There will be tradeoff between the choice of $\mathbf{w}$ and the number of mistakes along with corresponding certain penalty.

$$\dagger \min_{w,b} \mathbf{w}.\mathbf{w} + C \sum_j \xi_j, \text{ subject to } (\mathbf{w}.\mathbf{x_j} + b) \cdot y_j \geq 1 - \xi_j, \text{ where } \xi_j \geq 0, \forall j$$

Using Lagrange Multipliers and taking gradient leads to dual formation $(*)$ [5]. As known, the above is L^1-norm. For L^2-norm[5], it has the similar formula.

Not Linearly Separable If data is not linearly separable, an SVM classifier resorts kernel function to perform *nonlinear transformation* so as to map input feature space S to a higher-dimensional space: $\phi(\mathbf{x}) : S^n \to F$. However, the mapping could be computationally intensive. For example, polynomial transformation (d=2) function $\phi(\mathbf{x}) : \{x_1, x_2, ..., x_n\} \to \{x_n^2, ..., x_1^2, x_n x_{n-1}, ..., x_n x_1, ..., x_n, ...x_1, c\}$ which is a $\binom{n+2}{2} = (\frac{n^2+3n+2}{2})$-dimensional space. Relying on an essential observation that the algorithm only depends on the inner product of feature vectors, kernel functions could be used to compute $\phi(\mathbf{x_i}) \cdot \phi(\mathbf{x_j})$ on the original space as a kernel function has the following properties:

1. $K(\mathbf{x_i}, \mathbf{x_j})$ can be cheaply computed in the original space S
2. $K(\mathbf{x_i}, \mathbf{x_j}) = \phi(\mathbf{x_i}) \cdot \phi(\mathbf{x_j})$.

Instead of explicitly transforming the data from original space to the new space, the inner product of $\phi(\mathbf{x})$ could be cheaply computed in the original space. This is called *Kernel Trick*. The dual formula in terms of kernel function:

$$\dagger \max_\alpha \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j K(\mathbf{x_i}, \mathbf{x_j}) \text{ , where } K(\mathbf{x_i}, \mathbf{x_j}) = \phi(\mathbf{x_i}) \cdot \phi(\mathbf{x_j})$$

Four basic kernels:

- Polynomial of degree d: $K(\mathbf{x_i}, \mathbf{x_j}) = (\mathbf{x_i}^T \mathbf{x_j})^d$
- Polynomial of degree up to d: $K(\mathbf{x_i}, \mathbf{x_j}) = (\mathbf{x_i}^T \mathbf{x_j} + c)^d$, where $c > 0$
- Radial Basis Function (Gaussian) kernel: $K(\mathbf{x_i}, \mathbf{x_j}) = \exp\left(-\frac{||\mathbf{x_i} - \mathbf{x_j}||}{2\sigma^2}\right)$
- Sigmoid: $K(\mathbf{x_i}, \mathbf{x_j}) = tanh(\eta \mathbf{x_i^T} \mathbf{x_j} + \gamma)$

2 Decision Function: $f(\mathbf{x}) = sign(\mathbf{w} \cdot \Phi(x) + b)$

Solve dual formulation in the learning phase to obtain support vetoers α. At classification time: compute $\mathbf{w} \cdot \Phi(x) = \sum_i \alpha_i y_i K(\mathbf{x}, \mathbf{x_i}))$ and $b = y_k - \sum_i \alpha_i y_i K(\mathbf{x_k}, \mathbf{x_i}))$

Question: should all training data $\mathbf{x_i}$ be stored for future classification?

3 Discussion

1. Advantages: good generalization performance; automatic complexity control to reduce the overfitting; solve a variety problems with little tuning; a global optimum, not affected by local minima; do not suffer from the curse of dimensionality[4]
2. Hinge Loss: $\max(0, 1 - y_j \sum_i w_i x_i^j)$
3. If slack variables have a large penalty C, will it affect the accuracy?

4 Further Reading

[1] http://en.wikipedia.org/wiki/Kernel_trick
[2] http://en.wikipedia.org/wiki/Support_vector_machine
[3] http://www.csie.ntu.edu.tw/~cjlin/libsvm/
[4] http://en.wikipedia.org/wiki/Curse_of_dimensionality
[5] http://www.mathworks.com/help/stats/support-vector-machines-svm.html